대 한 민 국 특 허 청
KOREAN INDUSTRIAL PROPERTY OFFICE

별첨 사본은 아래 출원의 원본과 동일함을 증명함.

This is to certify that the following application annexed hereto is a true copy from the records of the Korean Industrial Property Office.

출 원 번 호 : 특허출원 2000년 제 42044 호
Application Number

출 원 년 월 일 : 2000년 07월 21일
Date of Application

출 원 인 : (주)씨앤에스 테크놀로지
Applicant(s)

2000 년 08 월 18 일

특 허 청
COMMISSIONER

【서류명】	특허출원서
【권리구분】	특허
【수신처】	특허청장
【참조번호】	0001
【제출일자】	2000.07.21
【발명의 명칭】	저전송율 영상통신을 위한 움직임 예측기 구조
【발명의 영문명칭】	MOTION ESTIMATOR ARCHITECTURE FOR LOW BIT RATE IMAGE COMMUNICATION
【출원인】	
【명칭】	(주)씨앤에스 테크놀로지
【출원인코드】	1-1998-096506-3
【대리인】	
【성명】	이종일
【대리인코드】	9-1998-000471-4
【포괄위임등록번호】	1999-065237-5
【대리인】	
【성명】	조희연
【대리인코드】	9-2000-000220-0
【포괄위임등록번호】	2000-040280-1
【발명자】	
【성명의 국문표기】	이영수
【성명의 영문표기】	LEE,Young Su
【주민등록번호】	690112-1037427
【우편번호】	135-010
【주소】	서울특별시 강남구 논현동 175-4 해주빌딩 6층
【국적】	KR
【심사청구】	청구
【취지】	특허법 제42조의 규정에 의한 출원, 특허법 제60조의 규정에 의한 출원심사 를 청구합니다. 대리인 이종일 (인) 대리인 조희연 (인)
【수수료】	
【기본출원료】	20 면 29,000 원
【가산출원료】	10 면 10,000 원
【우선권주장료】	0 건 0 원

【심사청구료】	7 항	333,000 원
【합계】	372,000 원	
【감면사유】	중소기업	
【감면후 수수료】	186,000 원	
【첨부서류】	1. 요약서·명세서(도면)_1통 2.중소기업법시행령 제2조에의한 중소기업에 해당함을 증명하는 서류 _3통[재무제표증명원사본,원천징수이행상황신고서확인원사본,사업자 등록증사본]	

【요약서】

【요약】

본 발명은 저전송율 영상 통신에 사용되는 적응형 움직임 예측기 구조에 관한 것이다. 본 발명의 목적은 1) 하드웨어적인 용량 크기를 줄이면서 적용하려는 영상의 특성 및 전송율에 맞는 움직임 예측기 구조를 구현토록 한다. 2) 저전송율 영상 특성이나 엔코더 성능에 적합한 서치방법을 선택적으로 적용하여 움직임 예측기의 성능을 최적화하도록 한다.

본 발명의 기술적 사상은 디램으로부터의 이전 서치윈도우 메모리 데이터와 움직임 벡터를 찾기 위한 현재 매크로 블럭 데이터를 각 데이터처리부(PE0 ~ PE8)의 처리 동작과 일치되도록 다중화시킨 후, 이전프레임 데이터와 현재프레임 데이터로 각 움직임 벡터의 MAE값을 비교 검출하여 움직임 벡터의 MAE값 중에서 가장 작은 MAE값을 갖는 움직임 벡터를 검출하도록 하는 저전송율 영상 통신에 사용되는 움직임 예측기 구조가 제시된다.

따라서, 본 발명은 적은 하드웨어 부담으로 엔코딩 효율을 높여야 하는 영상 전화 등에 적용할 수 있으며, 그 밖에 H.261/H.263, MPEG 표준을 따르는 모든 비디오 엔코더에 적용되어 사용될 수 있다.

【대표도】

도 4

【색인어】

움직임예측기, 이전프레임, 현재프레임, 정픽셀, 부픽셀, 엠펙(MPEG)

【명세서】

【발명의 명칭】

저전송율 영상통신을 위한 움직임 예측기 구조{MOTION ESTIMATOR ARCHITECTURE FOR LOW BIT RATE IMAGE COMMUNICATION}

【도면의 간단한 설명】

도 1은 현재 프레임의 매크로 블럭 개념을 설명하기 위한 개략도

도 2는 이전 프레임의 서치윈도우 및 움직임 벡터의 개념을 설명하기 위한 개략도

도 3은 도 2에 도시된 움직임 벡터 중에서 부픽셀 벡터의 위치를 나타낸 도면

도 4는 본 발명에 따른 움직임 예측기의 구조를 설명하기 위한 블럭 구성도

도 5a 및 도 5b는 본 발명의 부픽셀 서치과정을 설명하기 위한 도면

도 6a 및 도 6b는 본 발명에 따른 서치윈도우의 메모리 액세스방법을 설명하기 위한 도면

도 7a 및 7b는 본 발명에 따른 풀 서치 및 인터레이스 서치방법을 설명하기 위한 도면

도 8은 도 4에 도시된 제 1내지 제 5데이터처리부(PE0 ~ PE4)의 구조를 상세하게 나타낸 블럭구성도

도 9는 도 4에 도시된 제 6데이터처리부(PE5)의 구조를 상세하게 나타낸 블럭구성도

도 10은 도 4에 도시된 제 7내지 제 9데이터처리부(PE6 ~ PE8)의 구조를 상세하게 나타낸 블럭구성도

도 11은 도 4에 도시된 비교기의 구조를 나타낸 블럭구성도

<도면의 주요부분에 대한 부호의 설명>

100 : 이전 프레임저장부 200 : 현재 프레임저장부

300 : 멀티플렉서 400 : 데이터처리부

500 : 비교기 600 : 상태제어부

【발명의 상세한 설명】

【발명의 목적】

【발명이 속하는 기술분야 및 그 분야의 종래기술】

<16> 본 발명은 저전송율 영상 통신에 사용되는 적응형 움직임 예측기 구조에 관한 것이다. 특히, 하드웨어 크기를 줄이면서 적용하려는 영상의 특성 및 전송율에 맞는 움직임 예측기의 구조를 구현할 수 있는 기술에 관한 것이다.

<17> 일반적으로, 이미지 데이터의 압축 및 복원 기술은 멀티미디어 통신, 방송, 저장미디어 등의 각 분야에 있어서 필수적으로 사용되는 핵심 기술이다. 이러한 이미지 데이터의 압축, 복원 표준으로는 JPEG, MPEG, H.261/H.263 등이 있다. 이 중 H.261/H.263 은 저 전송율 영상 통신용으로 널리 사용되고 있다.

<18> 특히, 동영상의 압축은 공간적인 압축과 시간적인 압축으로 나뉘어지는데, 공간적인 압축 방법에 사용되는 알고리즘으로는 이산여현변환(Discrete Cosine Transform:DCT), 허프만 코딩(Huffman coding), 차분펄스부호변조(Differential Pulse Code Modulation:DPCM), 연속길이코딩(Run Length Coding:RLC)이 주로 사용되며, 시간적인 압축 방식으로는 움직임 예측(Motion estimation)방식이 사용된다.

<19> 움직임 예측방식(Motion estimation)은 이전(以前) 프레임과 현재(現在) 프레임의 시간적인 상관 관계에 의해 현재 프레임의 16×16 매크로 블럭을 이전 프레임의 주어진 서치영역 내에서 가장 차이가 적은 위치에 해당하는 움직임 벡터값을 구해 보냄으로써 압축율을 높이는 것으로, MPEG, H.261/H.263 표준을 적용한 모든 엔코더(encoder)에 사용되는 기술이다.

<20> 도 1은 현재 프레임 내의 매크로 블럭 개념을 설명하기 위한 개략도이다.

<21> 도 1에 도시된 바와 같이, 현재(current) 프레임 내의 매크로 블럭 크기는 MPEG, H.261/H.263 표준에 의해 16화소(pixel)×16화소(pixel)로 나타내며, 매크로 블럭(10)으로 정의된다.

<22> 이 때, 현재 프레임 크기가 352화소(pixel)×288화소(pixel)인 공통중간포맷(Common Interchange Format:CIF)의 경우 프레임당 396개의 매크로 블럭(10)이 존재하며, 396번의 움직임 예측(Motion estimation) 연산이 필요하다.

<23> 또한, 현재 프레임 크기가 176화소(pixel)×144화소(pixel)인 1/4CIF(Quarter CIF 이하:QCIF)의 경우 1프레임에 99번의 움직임 예측(Motion estimation) 연산이 필요하다는 것을 알 수 있다.

<24> 도 2는 서치윈도우 및 움직임 벡터의 개념을 설명하기 위한 개략도이다.

<25> 도 2에 도시된 바와 같이, 현재 프레임 내의 현재 매크로 블럭(10)은 이전(previous)프레임의 서치영역 상에서 화소 단위로 이동하며, 가장 차이가 적은 매크로 블럭을 찾게 된다.

<26> 이 때, 이전(previous) 프레임 상의 현재 매크로 블럭(10) 위치와 최상의 매치

(best match) 블럭(20) 위치의 이동 정도를 움직임 벡터(30)로 나타낸다.

<27> 움직임 예측(Motion estimator)의 계산량과 메모리 대역폭(bandwidth)은 가장 대표적인 움직임 예측방법인 풀(full) 서치방법을 사용했을 때 도 2에 도시된 바와 같이 움직임을 예측할 수 있다.

<28> 물론, 매크로 블럭(10)의 크기를 16화소×16 화소로 하고, 서치윈도우 거리를 8로 했을 때 가능한 벡터의 수는 X축으로 -8 ~ 8, Y축으로 -8 ~ 8 이며, 17화소×17화소의 289번 매크로 블럭의 비교가 필요하다.

<29> 이러한 289개의 움직임벡터 중 가장 유사한 매크로 블럭(10)의 벡터를 찾아내기 위해서는 주로 평균절대오차(Mean Absolute Error 이하:MAE)를 이용하며, 가장 적은 MAE에 해당되는 값을 움직임 벡터(30)로 결정한다.

<30> MAE는 현재(current)와 이전(previous) 화소값 차의 절대값을 더한 것으로 16화소×16화소값 차이의 절대값을 모두 더한 것이 된다.

<31> 즉, QCIF의 경우 모두 99개의 매크로 블럭이 있고, 각 매크로 블럭 당 MAE가 289번의 연산이 필요하기 때문에 1개의 프레임당 약 99×289(28,611)번의 MAE 연산이 필요하다.

<32> 실제로는 28,611번 보다 작으나, 프레임 경계면에서의 매크로 블럭은 서치영역이 제한되기 때문에 경계면을 고려하면 23,427번의 매크로 블럭 비교가 필요하게 된다.

<33> 상기와 같이 움직임 예측은 많은 계산량이 필요할 뿐만 아니라 매우 큰 메모리 대역폭(bandwidth)이 요구됨으로서 하드웨어적인 구현에 어려움이 있다.

<34> 즉, 이전 프레임 데이터와 현재 프레임 데이터는 일반적으로 디램(DRAM)에 보관 되

는데, 이는 프레임 데이터의 양이 무척 많기 때문이다. 그 예로써, CIF의 1 프레임 저장시 1메가(MEGA)비트 가량의 메모리가 요구된다.

<35> 또한, 매크로 블럭의 비교시 마다 디램(DRAM)으로부터 이전 프레임의 비교 매크로 블럭 데이터를 액세스하는 것은 매우 큰 메모리 대역폭을 필요로 한다.

<36> 도 2에 도시된 바와같이 QCIF, 이미지 포맷 그리고 초당 15프레임의 동영상의 경우 23,427×16×16×15 바이트/초(약 90M바이트/초)의 메모리 대역폭이 요구된다. 이처럼 큰 메모리 액세스의 요구 때문에 고속의 에스램(SRAM)을 캐쉬(cache) 메모리로 사용하는 경우가 많다.

<37> 도 3은 도 2에 도시된 움직임 벡터 중에서 부픽셀 포지션 및 그 수학식을 나타낸 도면이다.

<38> 도 3에 도시된 바와 같이, 대문자 A, B, C, D점은 정(integer) 픽셀포지션을 나타내고, 소문자 a, b, c, d점은 부(half) 픽셀포지션을 나타낸다. 즉, 움직임 벡터에는 정(integer) 픽셀벡터와 부(half) 픽셀 벡터가 존재함을 알 수 있다.

<39> 【수학식 1】

$$a = A$$

<40> $$b = (A + B + 1)/2$$

<41> $$c = (A + C + 1)/2$$

<42> $$d = (A + B + C + D + 2)/2$$

<43> 따라서, 상기와 같은 수학식 1에 의해 각 정 픽셀 및 부 픽셀포지션을 이용하여 부 픽셀 포지션의 움직임 예측을 계산할 수 있다.

【발명이 이루고자 하는 기술적 과제】

<44> 그러나, 상기와 같은 종래 움직임 예측 방식을 이용함으로써 다음과 같은 문제점이 발생된다.

<45> 첫째, 종래 움직임 예측에서는 많은 계산량을 필요로 하기 때문에 계산량을 줄이기 위해 계층(hierarchy) 서치, 3단계(3step) 서치, 서브 샘플링(sub sampling) 서치 등의 좀 더 빠른 서치 알고리즘이 발표 되었으나, 이러한 서치 방법은 하드웨어적인 실행에 어려움이 있어 대부분 풀(full) 서치 알고리즘을 사용하고 있다.

<46> 둘째, 풀 서치의 하드웨어적인 구조로는 시스토릭(systolic) 형태의 어레이 구조가 많이 사용되는데, 하드웨어적인 크기가 커지고 파이프 라인을 위한 레지스터의 양이 많기 때문에 동기 클럭(synchronous clock)의 로드(load)가 커지게 되는 단점이 있다.

<47> 셋째, 시스토릭(systolic) 형태의 어레이를 사용할 경우 부 픽셀 기능을 정 픽셀(integer pixel) 기능에 포함시키기 어렵기 때문에 별도의 부 픽셀 기능을 위한 하드웨어를 필요로 한다.

【발명의 구성 및 작용】

<48> 따라서, 본 발명은 상기한 문제점을 해결하기 위한 것으로써 본 발명의 목적은 하드웨어적인 용량 크기를 줄이면서 적용하려는 영상의 특성 및 전송율에 맞는 움직임 예측기 구조를 구현토록 하는 저전송율 영상통신을 위한 움직임 예측기 구조를 제공하는데 그 목적이 있다.

<49> 본 발명의 다른 목적은 저전송율 영상 특성이나 엔코더 성능에 적합한 서치방법을 선택적으로 적용하여 움직임 예측기의 성능을 최적화하도록 하는 저전송율 영상통신을

위한 움직임 예측기 구조를 제공하는데 그 목적이 있다.

<50> 상기한 본 발명의 목적을 달성하기 위한 기술적 사상은 디램으로부터의 이전 서치 윈도우 메모리 데이터와 움직임 벡터를 찾기 위한 현재 매크로 블럭 데이터를 각 데이터 처리부(PE0 ~ PE8) 동작과 일치되도록 다중화시킨 후, 이전프레임 데이터와 현재프레임 데이터로 각 움직임 벡터의 MAE값을 비교 검출하여 움직임 벡터의 MAE값 중에서 가장 작은 MAE값을 갖는 움직임 벡터를 검출하도록 한다.

<51> 이하, 본 발명의 실시예에 대한 구성 및 그 작용을 첨부한 도면을 참조하면서 상세히 설명하기로 한다.

<52> 도 4는 본 발명에 따른 움직임 예측기의 구조를 설명하기 위한 블럭구성도이다.

<53> 이전 프레임 데이터 중에서 현재 매크로 블럭의 서치윈도우 데이터가 저장되어 있는 이전 프레임저장부(100)와, 현재 프레임 데이터 중에서 움직임 벡터를 찾기 위한 현재 매크로 블럭 데이터가 저장되어 있는 현재 프레임저장부(200), 상기 이전 서치윈도우 데이터와 상기 현재 매크로블럭 데이터를 각 데이터처리부의 처리동작에 맞추어 다중화시키는 멀티플렉서(300), 상기 멀티플렉서(300)로부터 전송되는 상기 이전 프레임데이터와 현재 프레임데이터를 이용하여 각 움직임 벡터의 MAE값을 산출하는 데이터처리부(400), 상기 데이터처리부(400)으로부터 각 움직임 벡터의 MAE값을 비교 검출하여 움직임 벡터의 MAE값 중에서 가장 작은 MAE값을 갖는 움직임 벡터를 검출하는 비교기(500) 및 상기 각 구성요소의 동작 흐름을 총괄적으로 제어하는 상태제어부(600)를 포함하여 이루어져 있다.

<54> 이 때, 본 발명에서 인용되는 움직임 예측방식이란 동영상 프레임 간의 상관성을

이용하여 압축하기 위한 방법으로써 현재 프레임의 일정 블럭을 이전 프레임의 서치영역 상에서 움직임 정도를 찾은 후, 가장 차이가 적은 움직임 벡터와 화소 데이터의 차이값 만큼을 엔코딩하여 전송하는 기법을 말한다.

<55> 이전 프레임저장부(100)에는 이전 프레임 데이터 중에서 현재 매크로 블럭의 서치 윈도우 데이터가 저장되어 있으며, 메모리의 크기는 수학식 2에 도시된 바와 같다.

<56> 【수학식 2】

$$(PC_{mb} + (SD\times 2))^2 \times 8 \text{ bits}$$

<57> 이 때, PC_{mb} 은 매크로.블럭 화소 카운트이고, SD 는 서치 거리를 나타낸다.

<58> 즉, 16×16 매크로 블럭에 서치 거리를 8로 적용하면 $(16+(8\times2))^2\times8=1024\times8$비트 이다.

<59> 따라서, 움직임 예측의 실행을 고려하면, 메모리의 출력 데이터 폭은 32 비트가 적합하고, 메모리의 크기는 256×32 비트로 표기할 수 있다.

<60> 현재 프레임저장부(200)에는 현재 프레임 데이터 중에서 움직임 벡터를 찾기 위한 현재 매크로 블럭 데이터가 저장되어 있으며, 현재 매크로블럭의 메모리 크기는 수학식 3에 도시된 바와 같다.

<61> 【수학식 3】

$$PC_{mb}^2 \times 8 \text{ bits}$$

<62> 이 때, PC_{mb} 은 매크로 블럭 화소 카운트이며, 256×8비트가 된다.

<63> 즉, 데이터 폭을 32로 하면 64×32 비트 메모리로 표기할 수 있다.

<64> 또한, 멀티플렉서(300)로부터 전송되는 이전 프레임 데이터와 현재 프레임 데이터를 이용하여 각 움직임 벡터의 MAE값을 산출하는 데이터처리부(400)는 9개의 PE(Processing element)로 구성된다.

<65> 이는 한번에 9개의 움직임 벡터를 구하기 위한 것으로써, 9개의 PE를 사용한 데에는 다음과 같은 3가지의 이유 때문이다.

<66> 첫째, 칩 영역을 줄이기 위한 것이다. 시스토릭(systoric) 어레이 사용시에 256개의 PE를 필요로 하기 때문에 매우 큰 영역을 요구한다. 9개의 PE는 저 전송율 비디오 통신에 주로 사용되는 QCIF 이미지의 움직임 예측 뿐만 아니라 CIF 까지의 움직임 예측에 적합한 갯수이다.

<67> 둘째, 규칙(regular)적인 동작을 보장하기 위한 것이다. 서치 거리를 8로 할 때 X축의 벡터는 -8 ~ 8 까지 17가지가 된다. 왼쪽이나 오른쪽 보더(border)에서도 각각 -8 ~ 0, 0 ~ 8 까지 9개의 X축 벡터를 갖게 된다.

<68> 만약, PE를 8개로 사용하는 경우 한번에 8개의 벡터를 구한다면 9가지의 X축 벡터를 구하기 위하여 -8 ~ 0인 경우 -8 ~ -1 까지 한번 계산하고, 나머지 0에 대한 계산을 위해 한번 더 8개의 PE 연산을 수행해야 한다.

<69> 또한, 9개의 PE를 사용하여 -8 ~ 8까지 17개의 벡터에 대해 연산을 할 경우 -8 ~ 0까지 계산을 하고 다음에 0 ~ 8까지 0에 대해 중복을 하면 용이하게 제어하면서 효율적인 움직임 예측을 할 수 있다.

<70> 셋째, 부픽셀 벡터의 계산을 별도의 하드웨어 없이 9개의 PE를 이용하여 동작을 수행하기 위한 것이다.

<71> 즉, 도 3에 도시된 바와 같이, 부픽셀 포지션 d의 값을 구하기 위해서는 정픽셀 포지션 A, B, C, D 4개점의 값이 필요한데, 이 값을 가지고 부픽셀의 값을 계산할 수 있다.

<72> 도 5a 및 도 5b는 본 발명에 따른 부픽셀 서치방법을 설명하기 위한 도면이다.

<73> 도 5에 도시된 바와 같이, 부 픽셀 서치방법을 살펴보면 PE0 ~ PE4는 X,Y축 방향의 5개 부픽셀 값을 계산하고, PE5는 Y축 방향의 부픽셀 값을 계산하게 된다.

<74> 이 때, 이렇게 구해진 부픽셀 값은 PE6 ~ PE8에 입력되어 3개의 부픽셀 벡터값을 동시에 계산한다.

<75> 또한, 부픽셀 서치방법은 정픽셀 서치에서 구해진 움직임 벡터값을 포함한 주위 9개 부픽셀의 벡터에 대해 연산을 하므로써 PE를 9개 사용하게 되면 한번에 3개의 부픽셀 벡터값을 구할 수 있으므로 3번의 연산을 통해 부 픽셀 움직임 벡터값을 구할 수 있다.

<76> 도 6a 및 도 6b는 본 발명에 따른 서치 윈도우의 메모리 액세스방법을 설명하기 위한 도면이다.

<77> 도 6a, 도6b에 도시된 바와 같이, 외부 DRAM에 저장되어 있는 이전 이미지 데이터를 서치 윈도우(40a,40b)의 메모리로 가져올 때 효율적인 메모리 억세스 방법을 나타내고 있다.

<78> 현재 매크로 블럭에 대한 움직임 예측을 할 때, 외부 DRAM에서 서치 윈도우(40a) 메모리로 가져올 데이터 중 이전 매크로 블럭이 사용한 서치 윈도우(40b)에 절반이 겹쳐지는 것을 이용한 것이다.

<79> 즉, 매번 매크로 블럭에 대한 움직임 예측을 할 때 서치영역 전반에 대해 이전 이

미지 데이터를 가져오는 것이 아니라 이전 데이터와 겹치지 않는 새로운 데이터만 가져오는 것으로 매크로 블럭 번호가 짝수(even)인지 홀수(odd)인지에 따라 어드레스 계산을 다르게 하여 어드레스 콘트롤 문제를 해결 할 수 있다.

<80> 또한, 이 방법에 의해 외부 DRAM과의 억세스 타임을 줄일 수 있다. 본 발명에서 제시하는 움직임 예측은 영상의 특성과 인코더의 성능에 따라 풀 서치와 인터레이스(interlace) 서치를 선택할 수 있도록 설계가 되어 있다.

<81> 도 7a 및 7b는 본 발명에 따른 풀 서치 및 인터레이스 서치방법을 설명하기 위한 도면이다.

<82> 도7a, 도7b는 풀 서치와 인터레이스 서치 방법을 나타낸 것으로, 풀 서치 방법의 경우 모든 서치영역 내의 벡터값에 대해 움직임 예측을 수행하므로써 가장 정확한 움직임 벡터를 구할 수 있다.

<83> 그러나, 모든 벡터값에 대해 움직임 예측 벡터의 연산을 수행하므로 계산량이 많아져 움직임 예측의 속도가 엔코딩 속도를 결정하는 경우 오히려 엔코더 성능을 감소시킬 수 있다.

<84> 따라서, 영상 특성에 맞게 효율적으로 움직임 예측을 하여 엔코딩 프레임 비율을 증가시킬 필요가 있다. 움직임 예측의 계산량을 줄일 수 있는 패스트(fast) 서치 알고리즘이 있으나, 하드웨어적 구현상 어려움이 많고 풀서치 만큼 정확한 움직임 벡터를 찾지 못하는 단점이 있다.

<85> 본 발명에서의 움직임 예측 구조는 엔코딩 프레임 비율이 낮거나 QCIF 포맷 같은 작은 영상을 압축할 경우 풀 서치를 사용하여 보다 정확한 움직임 벡터를 찾을 수 있도

록 하고, 엔코딩 프레임 비율이 높으면서 CIF 포맷 이상의 영상을 압축할 경우에는 인터레이스 서치를 하여 빠르게 움직임 벡터를 찾을 수 있다.

<86> 풀 서치와 인터레이스 서치방법의 선택은 엔코더 성능에 의해 프로그램적으로 선택되도록 하였다. 여기서 빠르게 서치를 하기 위해 인터레이스 서치를 사용한 것은 풀 서치에 사용된 하드웨어적인 구조를 그대로 사용할 수 있기 때문이다. 즉, 움직임 벡터 서치의 정확성가 뛰어나기 때문이다.

<87> 또한, 움직임 벡터 서치의 정밀도를 높이기 위해 단순하게 인터레이스 서치방법만을 이용하는 것이 아니라 인터레이스 서치방법을 통해 움직임 벡터를 구한 뒤 Y축 방향으로 +/-1 씩 정 서치를 하여 다시 움직임 벡터값을 구하고 최종적으로 주변 9개의 부픽셀 벡터 서치에 의해 움직임 벡터를 구함으로써 풀 서치와 정밀도를 향상 시킬 수 있다.

<88> 도 8 내지 도 10은 도 4에 도시된 제 1 내지 제 9데이터처리부 즉, 프로세싱 엘리먼트(PE0 ~ PE8)의 구조를 상세하게 나타낸 블럭구성도이다.

<89> 먼저, 도 8에 도시된 제 1내지 제 5데이터처리부(PE0 ~ PE4)의 구조를 살펴보면, 현재의 픽셀포지션 데이터(CDATA) 및 이전의 픽셀포지션 데이터(PDATA)가 멀티플렉서(MUX)을 거쳐 동작모드지정신호(Bit-rate Allocation Signal 이하: BAS)에 의해 전송되는 4개의 현재 프로세싱 엘리멘트(PPE0 ~ PPE4)(400a,400b,400c,400d)와, 제 1가산기(ADD1, 410), 스케일러(Scaler, 420) 및 제 2가산기(430)를 포함하여 이루어져 있다.

<90> 즉, 제 1내지 제 5데이터처리부(PE0~PE4)는 메모리 버스폭이 32비트 이므로 한번에 4개의 화소를 메모리에서 불러 올 수 있으며, 여기에 맞추어 개개의 PE는 4화소를 계산하도록 설계되어 있다.

<91> 이 때, 프로세싱 엘리먼트(PE0 ~ PE4)의 기본동작은 4개의 이전 화소 포지션에서 상응하는 현재 화소 포지션의 화소 데이터 차를 구하여 그 값을 누산기(accumulator)에 저장하는 것이다.

<92> 이어서, 좀 더 구체적으로 구성 작용에 대하여 설명하면 다음과 같다.

<93> 먼저, MUX는 부픽셀 신호 형태에 따라 정픽셀(integer pel)과 부픽셀(half pel)로 구분된다. 정픽셀의 경우 MUX의 출력은 PDAT - CDAT가 되지만, 부픽셀의 경우는 이전 화소, PDAT 자체가 출력된다. 즉, PDAT 출력 또는 PDAT 값에 1을 더한 값 또는 0값이 출력된다.

<94> 이는 부픽셀 타입에 의해 결정되기 때문이다. 이러한 부픽셀 타입은 모두 4가지로 분류되며, 도 3에 도시된 부픽셀 포지션 a, b, c, d 점이 이에 해당한다.

<95> 예컨대, 부픽셀 타입이 a점일 경우 PPE0(400a)는 PDAT 값만 내보내고, PPE1~PPE3(400b,400c,400d)은 0 값을 내보내어 결국 차분신호(diff)로 a점의 부픽셀 값이 출력된다.

<96> 예컨대, 부픽셀 타입이 d점일 경우 PPE0(400a)는 PDAT + 1값, PPE1(400b)은 PDAT1, PPE2(400c)는 PDAT2, PPE3(400d)는 PDAT3 즉, +1 값을 출력한다. 이렇게 출력된 값들은 제 1가산기(ADD1,410)에서 더해지고 스케일러(scaler, 420)를 지나 부픽셀 값을 출력하게 된다.

<97> 또한, 정픽셀의 움직임 벡터 계산시 PE에서 계산된 MAE 값은 비교기(500)에서 비교하여 가장 작은 MAE값을 가진 벡터를 찾게 된다. 비교기(500)에서는 9개의 벡터에 대한 MAE가 결정된 후, 한번씩 동작하게 되며 상태제어부(600)에 의해 제어된다.

<98> 한편, 도 9 및 도 10에 도시된 제 6 내지 제 9데이터처리부 즉, PE5 및 PE6~ PE8의 구조는 도 8에 도시된 PE0 ~ PE4 구조와 유사하므로써 그 구성 요소에 대한 설명은 생략하기로 한다.

<99> 프로세싱 엘리먼트 PE5 및 PE6 ~ PE8가 정픽셀 모드일 경우에는 그 동작도 또한, PE0 ~ PE4 구조와 동일하다. 프로세싱 엘리멘트 즉, PE0 ~ PE4, PE5, PE6~PE8의 구조적 차이는 부픽셀 모드에서 차별화된다.

<100> 즉, 부픽셀 모드일 경우 PE0 ~ PE4에서는 4화소를 받아 하나의 부픽셀을 만들어 전송하며, PE5에서는 X축이 정픽셀인 화소값을 Y축 방향으로 각각 4화소씩 받아 4개의 부픽셀 값을 만들어 전송한다.

<101> 또한, PE6 ~ PE8에서는 PE0 ~ PE5를 통해 만들어진 부픽셀 값을 받아 정픽셀 모드일 경우와 같이 현재 화소 데이터와의 차이를 구하게 된다.

<102> 도 11은 도 4에 도시된 비교기의 구조를 나타낸 블럭구성도이다.

<103> 도 11에 도시된 비교기의 구조를 살펴보면, 데이터비교기(400)로부터 입력되는 9개의 평균절대오차(MAE0 ~ MAE8) 값을 다중화시키는 멀티플렉서(MUX,510)와, 상기 멀티플렉서(510)로부터 입력되는 MAE 값을 선별하여 선택적으로 전송하는 기판모듈(SUB,520) 및 상기 기판모듈(520)으로 부터 출력되는 MAE 값을 MAEP(previous MAE)값과 비교 검출하여 전송하는 비교부(530)를 포함하여 이루어져 있다.

<104> 좀 더 구체적으로 비교기의 작용에 대하여 살펴보면 다음과 같다.

<105> 먼저, 9개의 평균절대오차(MAE0 ~ MAE8)값 입력은 벡터 상태에 의해 하나씩 차례로 선택된다. 선택된 MAE는 기판(substraction)모듈(520)로 입력되며, 기판모듈(520)은

제로 벡터인 경우 제로벡터가중치(Zero Vector Weight:ZVW)값을 빼 주게 된다.

<106> 이는 벡터가 0인 경우 엔코딩 압축효율을 높일 수 있기 때문에 제로 벡터에 가중치 값을 주는 것이다. 비교부(530)에서는 MAE값을 MAEP(previous MAE)값과 비교하여 MAE값이 MAEP값보다 작을 경우 현재 MAE가 MAEP로 업데이트된다. 즉, 이 때의 벡터가 움직임 벡터로 업데이트된다.

【발명의 효과】

<107> 이상에서와 같이, 본 발명에 의한 저전송율 영상통신을 위한 움직임 예측기 구조에 따르면 다음과 같은 효과가 있다.

<108> 첫째, 저 전송율 영상통신에 사용될 수 있으며, 이전(previous) 서치 윈도우 메모리와 현재(current) 매크로 블럭 메모리의 채용으로 적은 메모리 대역폭(bandwidth) 상에서 동작이 가능하다. 또한, 메모리 어드레스 콘트롤을 통해 외부 DRAM으로부터 이전 서치 윈도우 메모리로의 데이터 억세스 타임을 줄일 수 있다.

<109> 둘째, 정픽셀, 부픽셀의 움직임 벡터를 같은 하드웨어로 계산할 수 있으며, 전체적으로 작은 하드웨어 상에서 성능을 향상시킬 수 있다.

<110> 셋째, 영상의 특성이나 엔코더 성능에 맞게 선택적으로 풀(full) 서치와 인터레이스(interlace) 서치를 적용하여 움직임 예측의 성능을 최적화시킬 수 있다.

<111> 넷째, 적은 하드웨어 부담으로 엔코딩 효율을 높여야 하는 영상 전화 등에 적용할 수 있으며, 그 밖에 H.261/H.263, MPEG 표준을 따르는 모든 비디오 엔코더에 적용되어 사용될 수 있다.

【특허청구범위】

【청구항 1】

이전 프레임 메모리에서 현재 매크로 블럭의 서치윈도우 데이터를 저장하는 이전 프레임저장수단과;

현재 프레임 메모리에서 움직임 벡터를 찾기 위한 현재 매크로 블럭 데이터를 저장하는 현재 프레임저장수단;

상기 이전 서치윈도우 데이터와 상기 현재 매크로 블럭 데이터를 각각의 데이터처리 동작과 일치되도록 다중화시키는 멀티플렉서;

복수개로 구성되어, 상기 멀티플렉서로부터 전송되는 상기 이전 서치윈도우 데이터와 현재 매크로 블럭 데이터를 이용하여 움직임 벡터의 평균절대오차(MAE)값을 산출하는 데이터처리수단;

상기 데이터처리수단으로부터 각 움직임 벡터의 MAE값을 비교 검출하여 상기 움직임 벡터의 MAE값 중에서 가장 작은 MAE값을 갖는 움직임 벡터를 검출하는 비교수단; 및

상기 각 구성수단의 동작 흐름을 총괄적으로 제어하는 상태제어수단을 포함하여 이루어진 것을 특징으로 하는 저전송율 영상통신을 위한 움직임 예측기 구조.

【청구항 2】

청구항 1에 있어서, 상기 움직임 벡터의 예측은 영상의 특성과 엔코더의 전송율에 따라 풀 서치방식 또는 인터레이스 서치방식 중에서 임의로 선택되는 하나의 방식을 사용하는 것을 특징으로 하는 저전송율 영상통신을 위한 움직임 예측기 구조.

【청구항 3】

청구항 1에 있어서, 상기 움직임 벡터의 검출은 X축,Y축 방향으로 -8 ~ +8 정픽셀 서치하여 정픽셀 움직임 벡터를 구한 뒤, 해당 움직임 벡터를 포함한 9개의 부픽셀을 서치하여 최종의 움직임 벡터값을 산출하는 것을 특징으로 하는 저전송율 영상통신을 위한 움직임 예측기 구조.

【청구항 4】

청구항 3에 있어서, 상기 부픽셀 서치방식은 9개의 PE를 사용하여 동시에 9개의 정픽셀 서치와 3개의 부픽셀 서치를 통해 부픽셀 움직임 벡터값을 산출하는 것을 특징으로 하는 저전송율 영상통신을 위한 움직임 예측기 구조.

【청구항 5】

청구항 3에 있어서, 상기 움직임 벡터의 검출은 정픽셀 서치방식과 부픽셀 서치방식을 동시에 사용하는 것을 특징으로 하는 저전송율 영상통신을 위한 움직임 예측기 구조.

【청구항 6】

청구항 1에 있어서, 상기 이전 프레임저장수단에서 현재 매크로 블럭의 서치윈도우 데이터를 가져올때 이전 서치윈도우 데이터와 겹치지 않는 새로운 데이터만을 업데이트 하는 것을 특징으로 하는 저전송율 영상통신을 위한 움직임 예측기 구조.

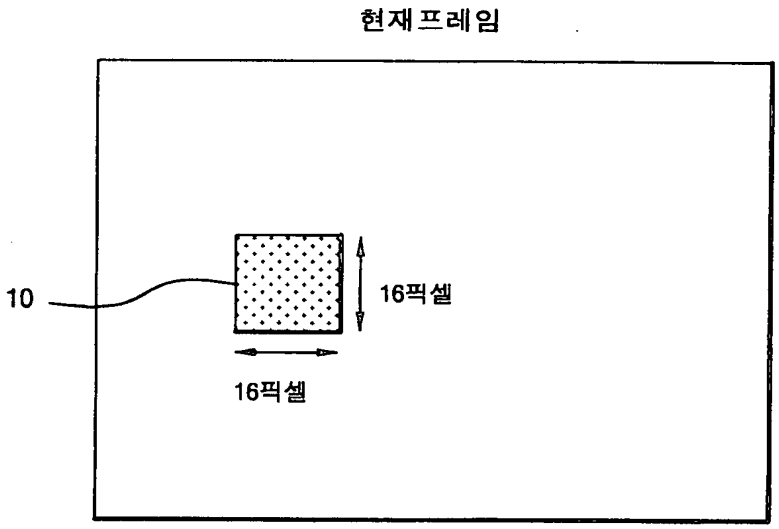
【청구항 7】

청구항 6에 있어서, 상기 현재 매크로 블럭에 대한 움직임 예측을 할 때 업데이트

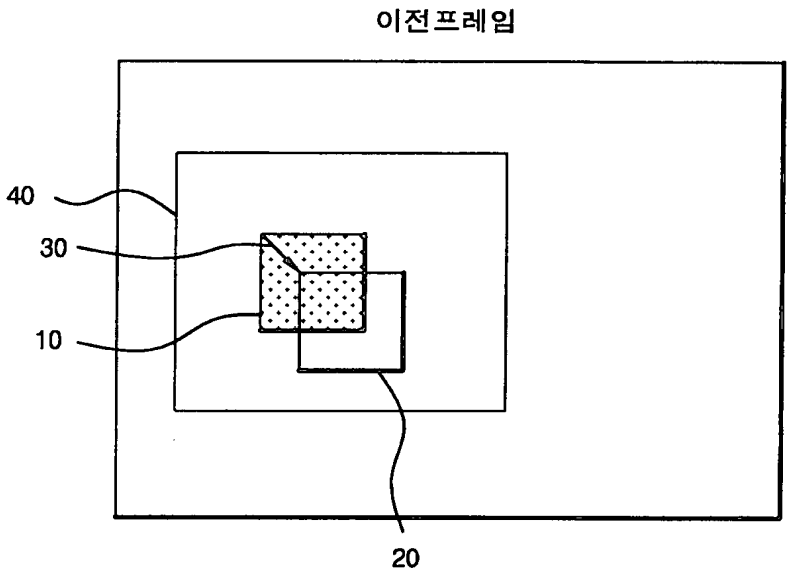
한 매크로 블럭 번호가 짝수(even) 또는 홀수(odd)에 따라 어드레스 계산을 다르게 하는 것을 특징으로 하는 저전송율 영상통신을 위한 움직임 예측기 구조.

【도면】

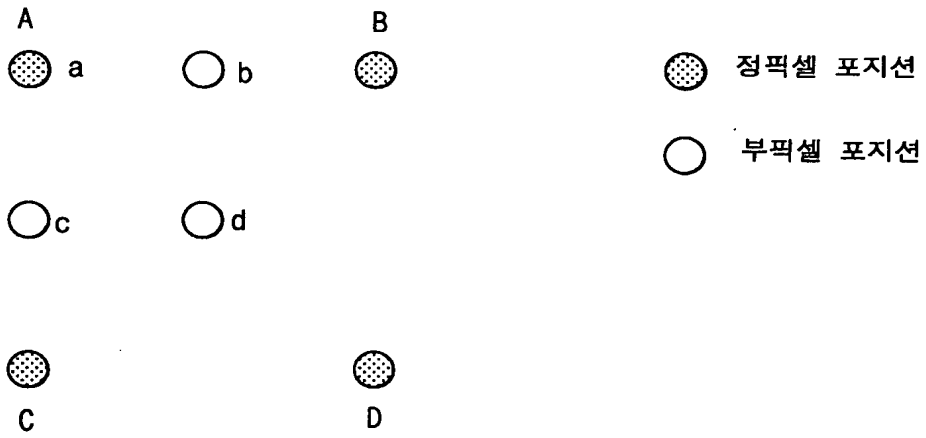
【도 1】



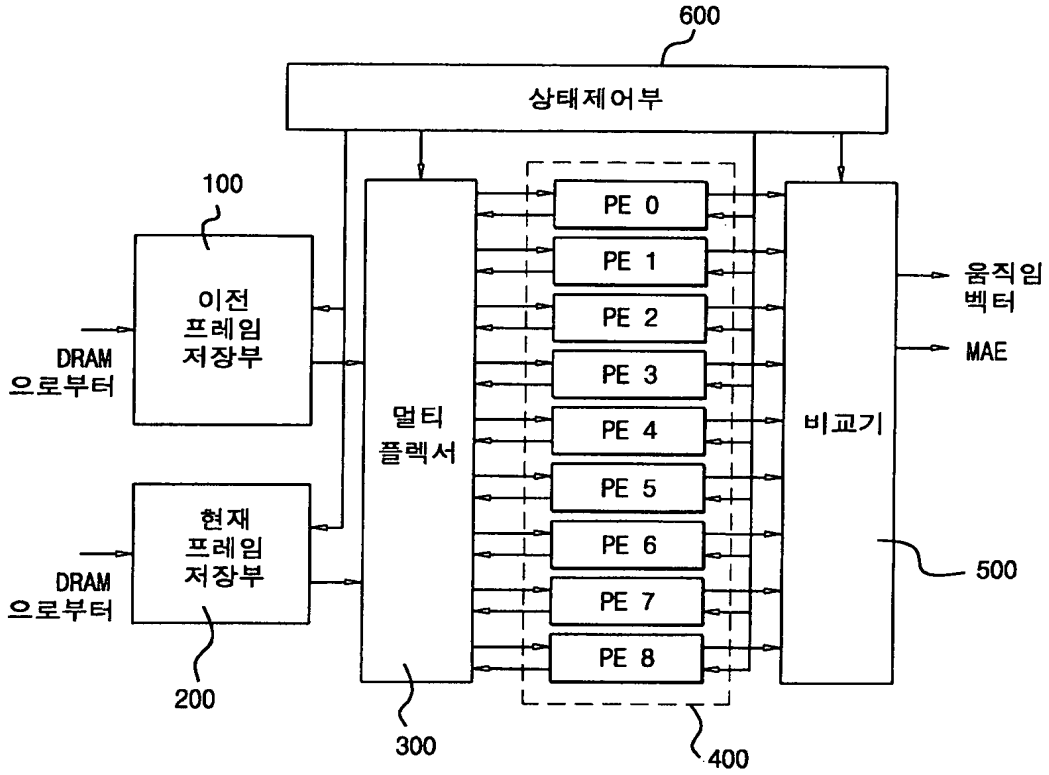
【도 2】



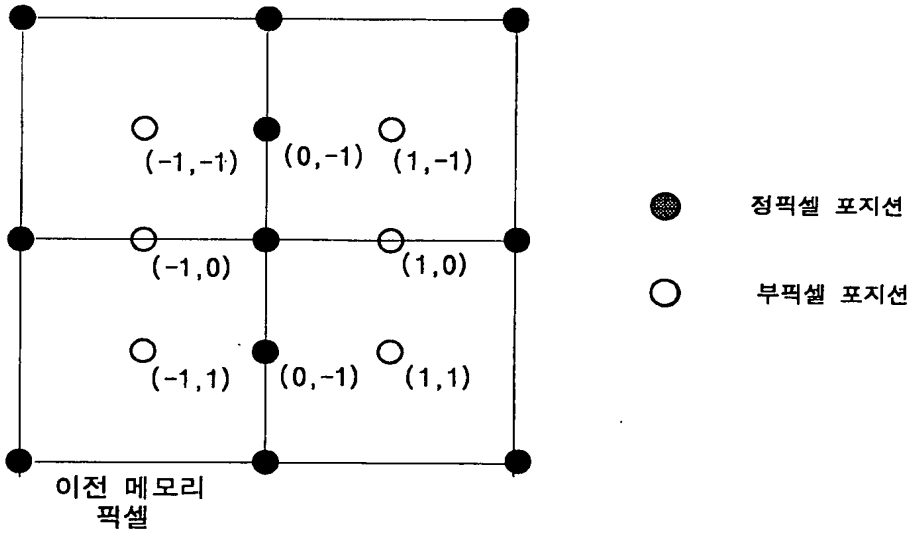
【도 3】


A
B
a
b
정픽셀 포지션
부픽셀 포지션
c
d
C
D

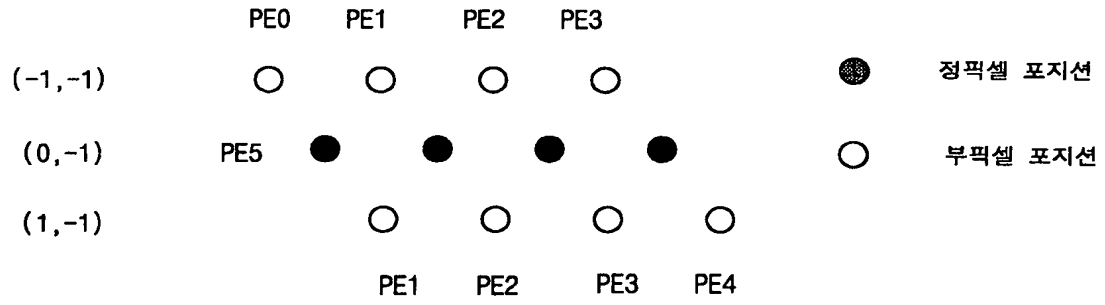
【도 4】


600
상태제어부
100
이전
프레임
저장부
DRAM
으로부터
현재
프레임
저장부
DRAM
으로부터
200
멀티
플렉서
300
PE 0
PE 1
PE 2
PE 3
PE 4
PE 5
PE 6
PE 7
PE 8
400
비교기
500
움직임
벡터
MAE

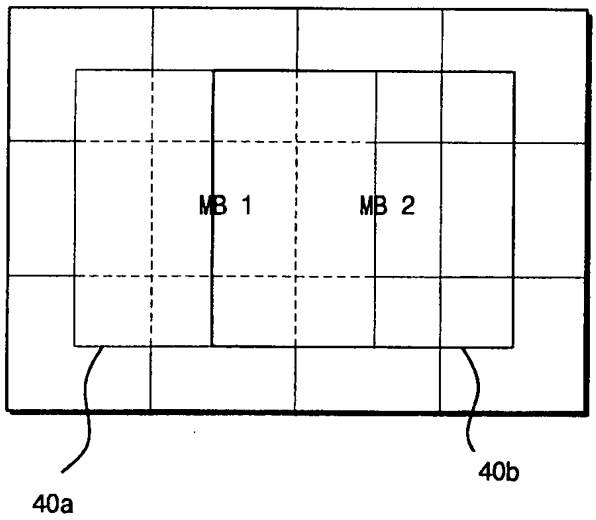
【도 5a】


(-1,-1)
(0,-1)
(1,-1)
(-1,0)
(1,0)
(-1,1)
(0,-1)
(1,1)
이전 메모리 픽셀
정픽셀 포지션
부픽셀 포지션

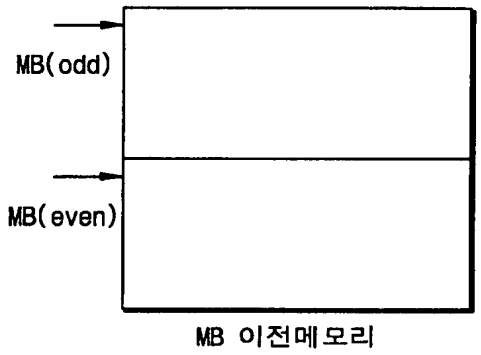
【도 5b】


PE0
PE1
PE2
PE3
(-1,-1)
(0,-1)
PE5
(1,-1)
PE1
PE2
PE3
PE4
정픽셀 포지션
부픽셀 포지션

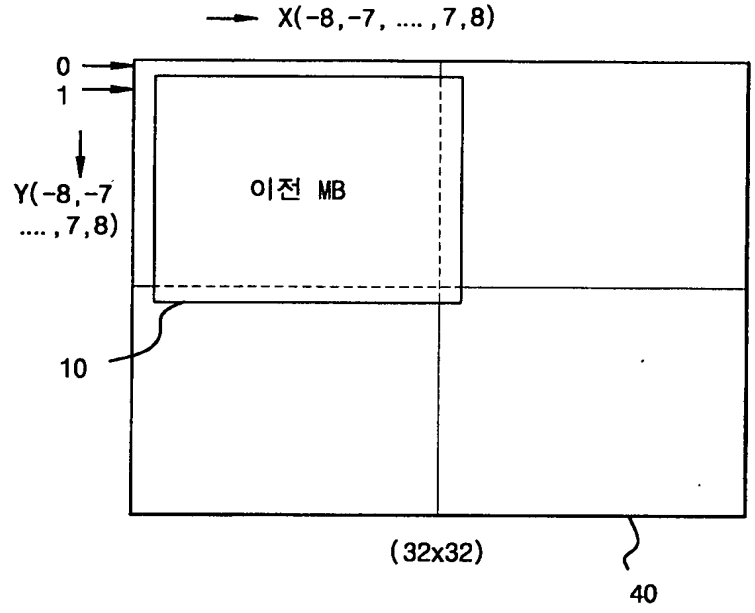
【도 6a】


MB 1
MB 2
40a
40b

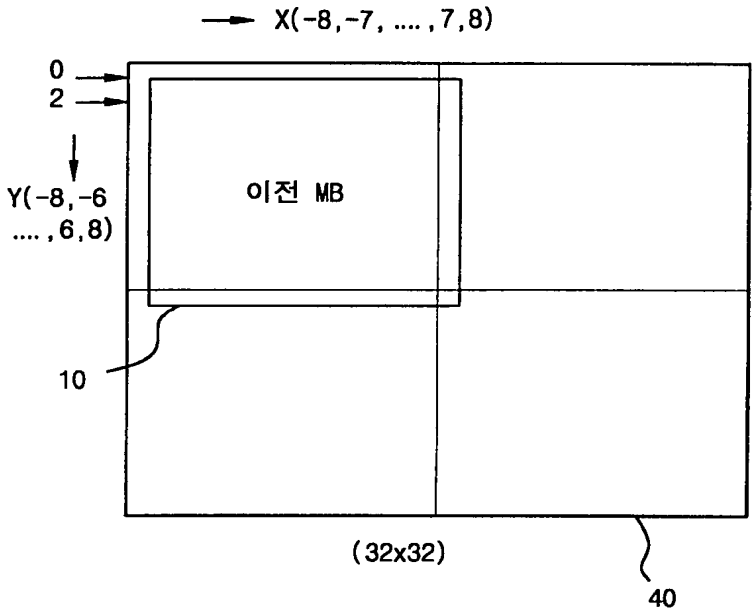
【도 6b】



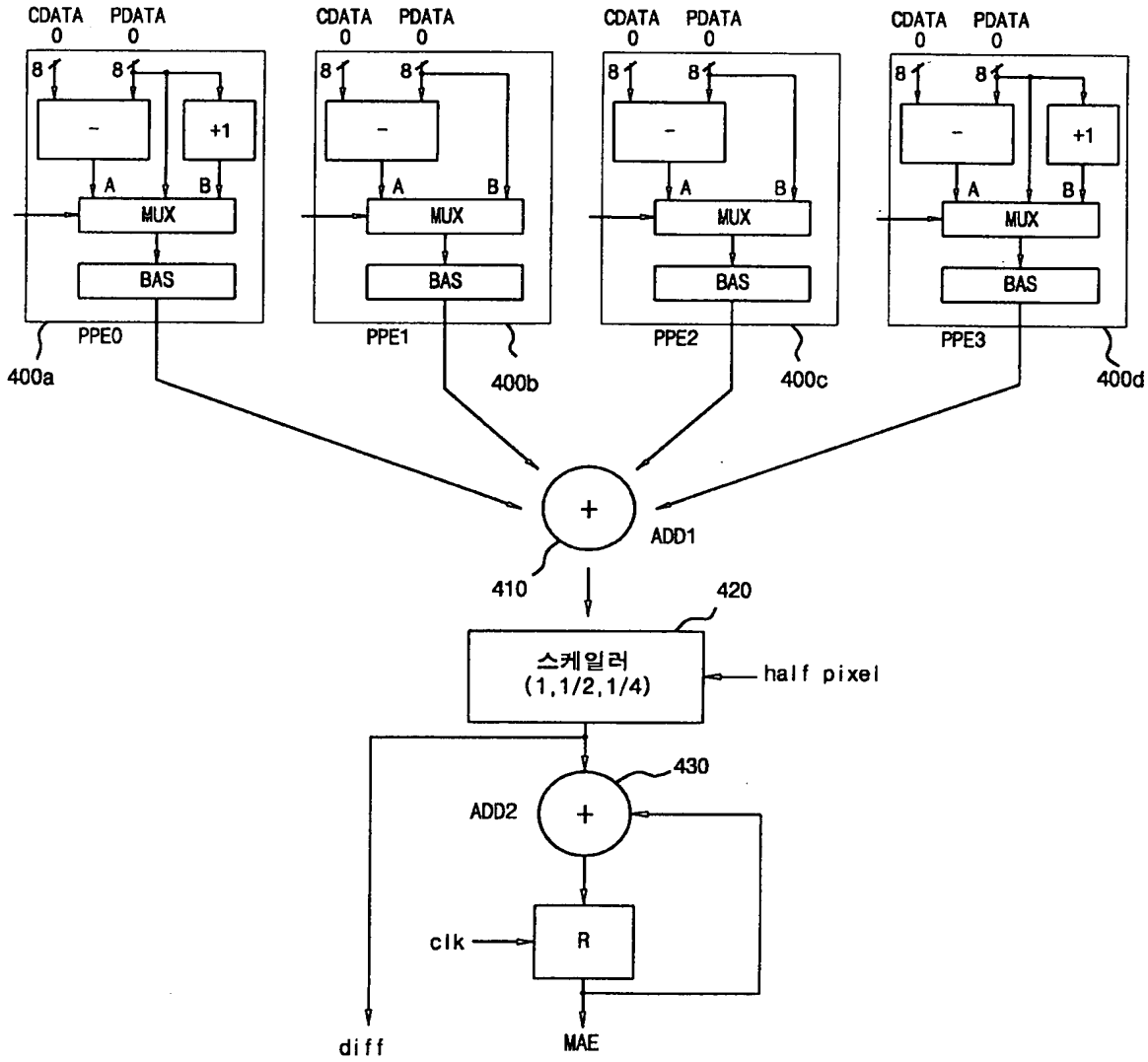
【도 7a】



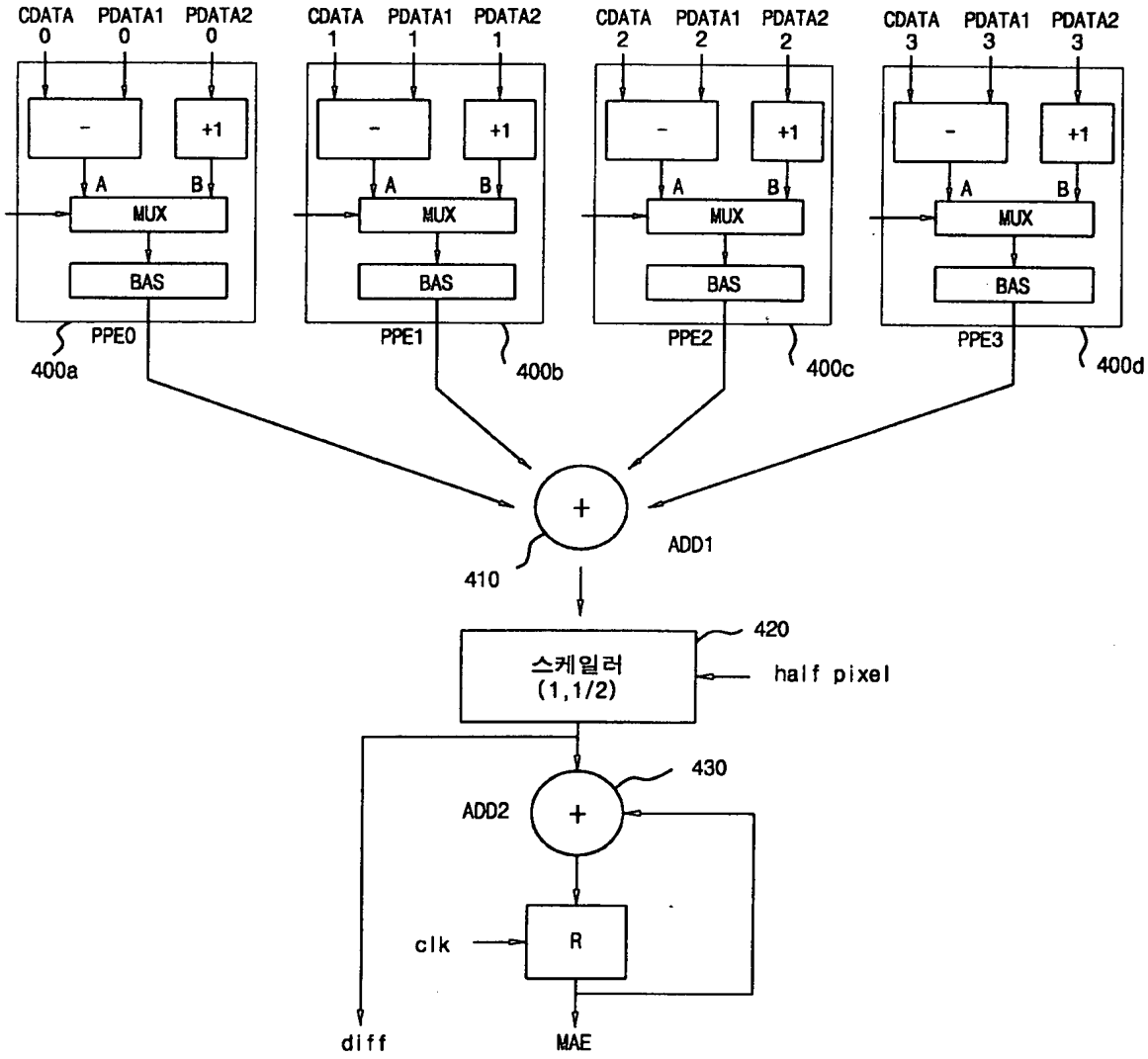
【도 7b】



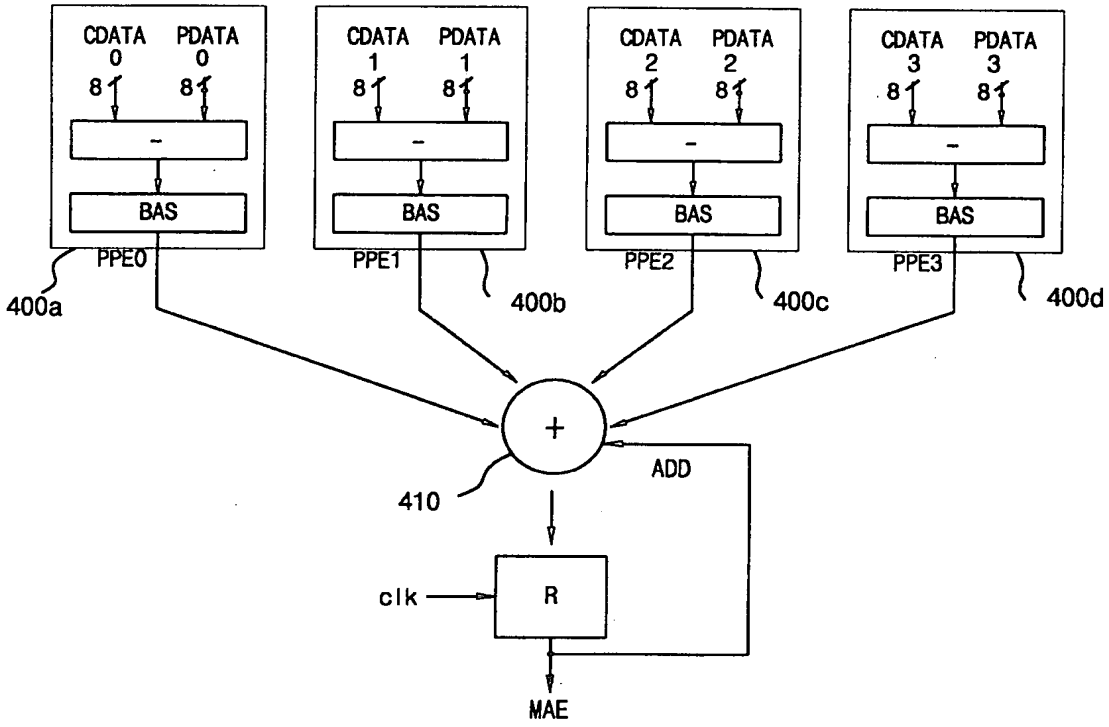
【도 8】


CDATA
PDATA
0
0
8
8
-
+1
A
B
MUX
BAS
PPE0
PPE1
PPE2
PPE3
400a
400b
400c
400d
+
ADD1
410
420
스케일러
(1,1/2,1/4)
half pixel
430
ADD2
clk
R
diff
MAE

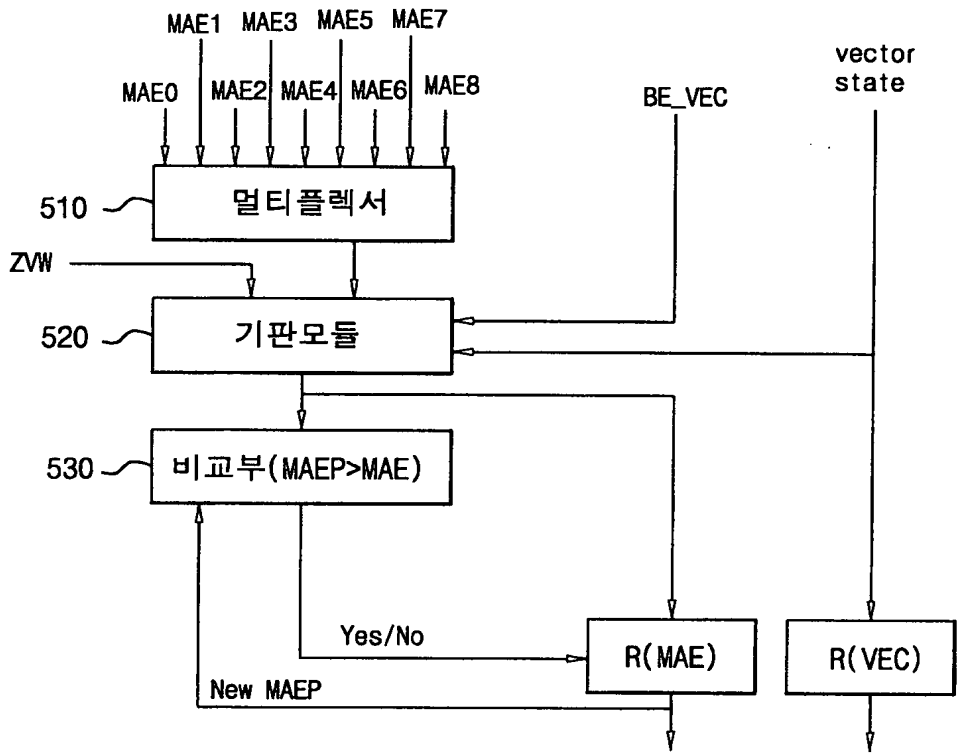
【도 9】


CDATA 0
PDATA1 0
PDATA2 0
CDATA 1
PDATA1 1
PDATA2 1
CDATA 2
PDATA1 2
PDATA2 2
CDATA 3
PDATA1 3
PDATA2 3
-
+1
A
B
MUX
BAS
PPE0
PPE1
PPE2
PPE3
400a
400b
400c
400d
+
ADD1
410
420
스케일러
(1,1/2)
half pixel
430
ADD2
clk
R
diff
MAE

【도 10】


CDATA 0
PDATA 0
8
8
-
BAS
PPE0
400a
CDATA 1
PDATA 1
PPE1
400b
CDATA 2
PDATA 2
PPE2
400c
CDATA 3
PDATA 3
PPE3
400d
+
ADD
410
clk
R
MAE

【도 11】


MAE1
MAE3
MAE5
MAE7
MAE0
MAE2
MAE4
MAE6
MAE8
BE_VEC
vector state
510
멀티플렉서
ZVW
520
기판모듈
530
비교부(MAEP>MAE)
Yes/No
New MAEP
R(MAE)
R(VEC)